Klasifikasi Deteksi Tautan Menggunakan *Random Forest* dan Metode Seleksi Fitur Berbasis Website

Lidya Marcelena[1], Anita Qoiriah[2]

[1, 2] Teknik Informatika, Universitas Negeri Surabaya
[1]lidyamarcelena.22012@unesa.ac.id
[2]anitaqoiriah@unesa.ac.id

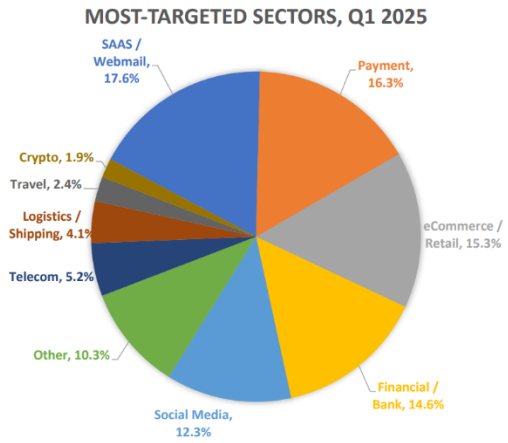
***Abstrak* — *Phishing* merupakan salah satu bentuk kejahatan siber yang bertujuan mencuri informasi pribadi pengguna melalui tautan palsu atau tiruan. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi deteksi tautan *phishing* dan *legitimate* menggunakan algoritma *machine learning* dan metode seleksi fitur berbasis website. Algoritma yang digunakan dalam penelitian ini adalah algoritma *Random Forest* dengan metode seleksi fitur *Chi-Square*. Proses penelitian meliputi pra pemrosesan data, ekstraksi fitur berdasarkan struktur dan konten website dalam sebuah tautan, seleksi fitur, *hyperparameter tuning*, serta evaluasi model klasifikasi. Kinerja model dievaluasi menggunakan beberapa metrik pengujian untuk menentukan model terbaik dalam mendeteksi tautan *phishing*. Hasil penelitian menunjukkan bahwa algoritma *machine learning* dan metode seleksi fitur mampu digunakan untuk mendeteksi tautan *phishing* secara efektif. Model terbaik kemudian diimplementasikan ke dalam sistem berbasis website sehingga dapat membantu pengguna mengidentifikasi tautan berbahaya secara *realtime*. Sistem yang dikembangkan diharapkan dapat meningkatkan keamanan pengguna saat mengakses internet serta menjadi solusi pendukung dalam upaya pencegahan serangan *phishing* yang semakin berkembang.**

***Kata Kunci* — *Phishing. Machine Learning, Random Forest, Logistic Regression, Chi-Square*.**

I. PENDAHULUAN

Di era perkembangan teknologi saat ini, masyarakat tidak bisa lepas dari internet. Internet adalah salah satu infrastruktur jaringan terbesar di dunia yang memfasilitasi berbagai informasi secara daring. Sebagai sarana media informasi global, internet selalu memberikan informasi yang *up-to-date* dengan kondisi yang *trend* di momen tertentu. Namun, di sisi lain, internet juga dapat digunakan sebagai salah satu jalur kejahatan oleh para *cybercrime*. Salah satu bentuk kejahatan yang paling popular ditemui adalah serangan phishing [1].

Phishing merupakan sebuah penipuan yang dilakukan secara *online*. Pelaku yang melakukan *phishing* kerap dikenal dengan sebutan *phisher*. Seorang *phisher* biasanya akan melakukan berbagai cara penipuan *online* untuk mendapatkan informasi pribadi (nama lengkap, tempat dan tanggal lahir, alamat rumah) atau informasi keuangan (*email*, kata sandi, *password*, nomer rekening, sandi rekening) milik korban. Penipuan yang marak terjadi saat ini adalah *phishing* website. *Phishing* website merupakan website tiruan yang dibuat semirip mungkin dengan website aslinya agar bisa mengelabui pengguna [1].



Gbr. 1 Grafik *Anti Phishing Working Group* Kuartal 1 2025

Menurut grafik dari situs *Anti-Phising Working Group* (APWG) pada kuartal pertama tahun 2025 serangan *online* yang sering menjadi target *phishing* adalah *webmail* (17,6%), *payment* (16,3%), *e-commerce* (15,3%), *financial/bank* (14,6%), dan *social media* (12,3%). Dari lima sektor di atas dapat disimpulkan bahwa layanan berbasis internet dan keuangan merupakan target utama dalam serangan *phishing* karena tingginya nilai data dan akses yang dimiliki pengguna pada platform-platform tersebut.

Phishing tidak hanya menimbulkan kerugian finansial bagi individu dan organisasi, tetapi juga dapat merusak reputasi dan kepercayaan publik. Selain itu, serangan ini dapat menjadi pintu masuk bagi para *cybercrime* untuk melanjutkan tindak kejahatannya dengan lebih serius, seperti *ransomware* atau pengambil alihan akun untuk disalahgunakan [2].

Machine learning merupakan salah satu cabang dari *artificial intelligence* yang memungkinkan sebuah sistem untuk belajar secara otomatis dari data yang diberikan, serta meningkatkan kemampuannya seiring berjalannya waktu tanpa harus diprogram secara langsung. Algoritma dalam *machine learning* dapat mengenali pola-pola dalam data dan mempelajarinya guna menghasilkan prediksi secara mandiri. Secara sederhana, model dan algoritma pembelajaran mesin memperoleh pengetahuan melalui proses pengalaman [3].

Random Forest merupakan algoritma yang popular dan terbukti efektif untuk digunakan. *Random Forest* memiliki beberapa keunggulan, diantaranya bisa mengolah data dalam dimensi besar, tahan terhadap *overfitting*, dan bisa memberikan informasi penting terkait fitur-fitur yang digunakan [4].

Untuk mendukung kinerja *Random Forest*, penelitian ini menggunakan *Chi-Square* sebagai metode seleksi fitur. *Chi-Square* berfungsi untuk mengukur tingkat ketergantungan antara fitur dengan label kelas, sehingga dapat membantu memilih fitur-fitur yang paling berpengaruh terhadap hasil klasifikasi [5].

Pada penelitian ini, penulis mengembangkan sistem klasifikasi deteksi tautan *phishing* menggunakan algoritma *Random Forest* dengan metode seleksi fitur *Chi-Square* berbasis website. Penelitian ini bertujuan membandingkan kinerja algoritma, baik dengan seleksi fitur maupun tanpa seleksi fitur, untuk memperoleh model terbaik. Model dengan kinerja terbaik kemudian diimplementasikan ke dalam website deteksi tautan *phishing* dan *legitimate* guna membantu pengguna mengidentifikasi keamanan tautan sebelum diakses serta mengurangi risiko menjadi korban *phishing*.

II. TINJAUAN PUSTAKA

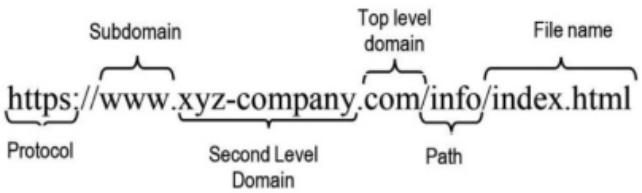
Bab ini membahas teori-teori yang menjadi landasan dalam penelitian serta konsep-konsep yang mendukung proses pengembangan model dan implementasi sistem.

A. Phishing

Phishing adalah bentuk penipuan *online* yang bertujuan mencuri data pribadi milik korban, hal tersebut bisa meliputi: informasi pribadi, kata sandi, dan beberapa hal krusial lainnya. Aksi ini biasanya dilakukan untuk menyalahgunakan identitas, menguras dana, atau merusak reputasi seseorang. Pelaku *phishing (phisher)* sering menyamar sebagai pihak resmi, misalnya perusahaan atau lembaga keuangan, guna meyakinkan korban agar mau menyerahkan data sensitif melalui tipu daya. Modus yang digunakan biasanya mengandalkan rekayasa sosial, manipulasi psikologis, serta teknologi, sehingga dapat merugikan individu maupun organisasi [6].

B. Uniform Resource Locator

Uniform Resource Locator (URL) adalah alamat yang digunakan untuk mengakses sumber daya di internet. Setiap URL memiliki struktur dan format tertentu yang umumnya terdiri dari penanda protokol untuk menentukan jenis protokol yang digunakan serta nama sumber daya berupa alamat IP atau domain. Dalam *phishing* website, struktur ini sering dimanipulasi oleh para *cybercrime* dengan cara mengubah salah satu komponennya untuk menipu pengguna dan menyebarkan tautan berbahaya yang merugikan korban [7].



Gbr. 2 Struktur Umum *Uniform Resource Locator*

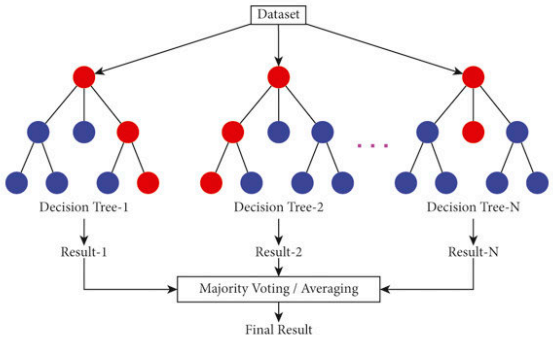
Struktur URL umumnya diawali dengan protokol yang digunakan untuk mengakses halaman web. Setelah itu, URL terdiri atas *subdomain* dan *second-level domain* (SLD) yang biasanya mengacu pada identitas organisasi pada server hosting. Selanjutnya, URL diakhiri dengan *top-level domain* (TLD) yang menunjukkan klasifikasi domain serta posisinya dalam *root* DNS internet [7].

C. Klasifikasi

Klasifikasi merupakan salah satu teknik dalam *supervised learning* yang digunakan untuk menganalisis data latih *(training dataset)* dan membangun model guna memprediksi kelas dari suatu objek yang belum diketahui. Pada proses klasifikasi, terdapat dua jenis data yang digunakan, yaitu data latih yang telah diberi label untuk melatih model, serta data uji *(testing dataset)* yang tidak berlabel dan dipakai untuk memprediksi label serta mengukur akurasi model. Umumnya, proporsi data latih lebih besar dibandingkan data uji, dengan pembagian yang sering digunakan berada pada kisaran 70–90% untuk data latih [8].

D. Random Forest

Random Forest adalah algoritma machine learning yang tergolong dalam *ensemble learning*, yakni teknik yang memanfaatkan gabungan beberapa model untuk memperoleh hasil prediksi yang lebih akurat dan stabil. Algoritma ini membangun banyak pohon keputusan yang masing-masing dilatih secara independen, lalu mengombinasikan hasil prediksi dari setiap pohon, umumnya dengan perhitungan rata-rata, sehingga performa model menjadi lebih optimal [2].



Gbr. 3 Algoritma Random Forest

Random Forest merupakan gabungan dari banyak pohon keputusan *(decision tree),* semakin banyak jumlah pohon yang dibangun maka semakin tinggi pula akurasi hasil prediksi yang diperoleh. Melalui mekanisme tersebut, *Random Forest* mampu meminimalkan risiko *overfitting* yang sering terjadi pada *decision tree* tunggal sekaligus menghasilkan model yang lebih stabil dan konsisten dalam melakukan klasifikasi [2].

E. Chi-Square

Chi-Square merupakan metode seleksi fitur berbasis uji statistik yang digunakan untuk mengukur tingkat keterkaitan

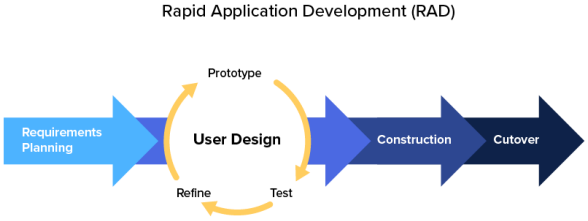
antara suatu fitur dan kelas target. Metode ini membandingkan frekuensi kemunculan fitur yang diamati dengan frekuensi yang diharapkan apabila tidak terdapat hubungan antara fitur dan kategori kelas. Semakin besar nilai *Chi-Square* yang dihasilkan, semakin tinggi tingkat relevansi fitur tersebut dalam membedakan kelas, sehingga fitur dengan nilai rendah dapat dieliminasi untuk meningkatkan efektivitas dan efisiensi proses klasifikasi [9].

F. Website

Website adalah salah satu media informasi dan promosi alternatif yang digunakan untuk mencari serta menyebarkan informasi, sekaligus berfungsi sebagai sarana pemasaran suatu perusahaan yang dapat diakses oleh siapa pun di seluruh dunia. Selain itu, website juga berperan sebagai identitas digital yang mencerminkan citra, kredibilitas, dan profesionalitas suatu individu, organisasi, maupun perusahaan di dunia maya. Website tersusun dari sekumpulan halaman dalam satu domain yang memuat berbagai jenis konten yang dapat diakses dan dilihat oleh pengguna internet melalui mesin pencari. Kontennya bisa berupa teks, gambar, video, ilustrasi, dan berbagai informasi lainnya [10].

G. Rapid Application Development

Rapid Application Development (RAD) merupakan model pengembangan perangkat lunak yang bersifat inkremental dan dapat dirancang dengan waktu pengerjaan yang singkat. Model ini menekankan pada siklus pengembangan yang cepat dan iteratif, di mana sistem dibangun melalui pembuatan komponen-komponen perangkat lunak secara modular. RAD dapat dianggap sebagai bentuk adaptasi dari metode *Waterfall* namun dengan pendekatan yang lebih fleksibel dan berorientasi pada kecepatan konstruksi serta keterlibatan pengguna dalam setiap tahap pengembangannya [11].



Gbr. 4 *Rapid Application Development*

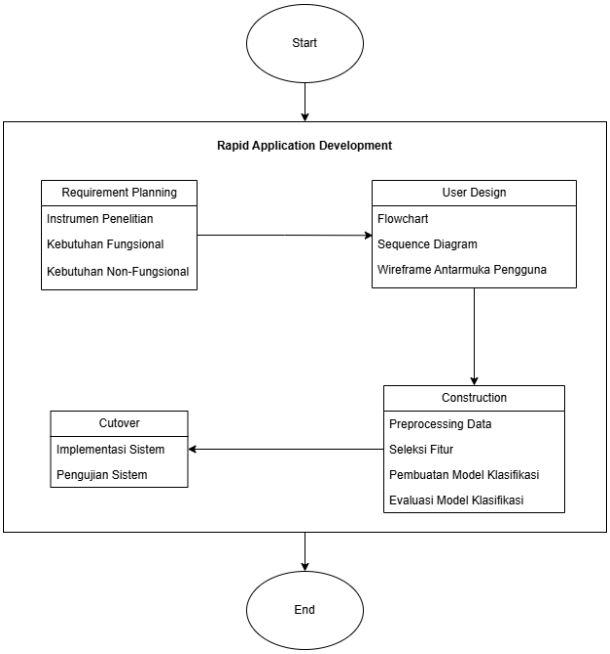
Metode RAD memiliki empat tahap utama, yaitu: *requirement planning, user design, construction, dan cutover*. Tahap *requirement planning* dilakukan dengan menganalisis kebutuhan fungsional dan non-fungsional sistem. Tahap *user design* berfokus pada antarmuka sistem dan alur kerja sistem. Tahap *construction* merupakan proses implementasi hasil desain ke dalam bentuk sistem yang dapat dijalankan. Tahap terakhir yaitu *cutover* dilakukan dengan pengujian sistem guna memastikan semua fungsi berjalan sesuai kebutuhan pengguna [12].

H. Blackbox Testing

Blackbox testing adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsi-fungsi sistem tanpa memperhatikan struktur internal atau kode program. Pengujian ini dilakukan dengan cara memberi *input* dan memeriksa kesesuaian *output* yang dihasilkan terhadap kebutuhan atau spesifikasi sistem. *Blackbox testing* digunakan untuk memastikan fitur dan alur proses aplikasi berjalan sesuai harapan pengguna, meminimalisir kesalahan, dan memastikan seluruh fungsi bekerja sebagaimana mestinya berdasarkan spesifikasi perangkat lunak [13].

III. METODOLOGI PENELITIAN

Penelitian ini akan menghasilkan *output* yaitu sistem deteksi tautan berbasis website dan akan menggunakan salah satu metode dari SDLC *(Software Development Life Cycle)* yaitu RAD *(Rapid Application Development)*. Alur detail penelitian dapat dilihat pada Gbr. 5.



Gbr. 5 Alur Penelitian

A. Requirement Planning

Tahap *requirement planning* adalah proses awal yang melibatkan peneliti untuk mengidentifikasi tujuan sistem serta menentukan kebutuhan informasi yang diperlukan agar sistem dapat berjalan sesuai fungsinya [11]. Pada penelitian ini, tahap *requirement planning* akan dibagi menjadi 3, yaitu: analisis kebutuhan fungsional, analisis kebutuhan non-fungsional, analisis kebutuhan perangkat penelitian.

B. User Design

Tahap *user design* bertujuan untuk merancang solusi yang sesuai dengan kebutuhan pengguna, memastikan sistem berjalan sesuai harapan, serta mampu mengatasi permasalahan

yang ada [14]. Pada penelitian ini, tahap *user design* akan dibagi menjadi 2, yaitu: perancangan desain antarmuka dan perancangan alur sistem.

C. Construction

Tahap *construction* merupakan proses membangun sistem berdasarkan rancangan yang telah disusun sebelumnya [15]. Tahap *construction* pada penelitian ini dimulai dari pra pemrosesan data, seleksi fitur, dan pembuatan model klasifikasi, dan pembuatan website.

D. Cutover

Tahap *cutover* merupakan tahap pengujian menyeluruh terhadap sistem yang telah dikembangkan [15]. Tahap ini meliputi proses pengujian menggunakan metode *Blackbox Testing* untuk memastikan bahwa sistem telah berfungsi sesuai kebutuhan pengguna.

IV. HASIL DAN PEMBAHASAN

Hasil dan pembahasan pada penelitian ini akan dilakukan proses implementasi serta pengujian terhadap sistem yang telah dirancang. Hasil dari setiap tahapan pengolahan data dan proses pengujian kemudian dianalisis untuk mengetahui kinerja sistem dalam melakukan proses klasifikasi.

A. Requirement Planning

Pada tahap *requirement planning*, dilakukan identifikasi dan perumusan kebutuhan sistem yang akan digunakan dalam penelitian. Berikut hasil identifikasi dari tahap ini:

1) Hasil Analisis Kebutuhan Fungsional: Kebutuhan fungsional pada penelitian ini mencakup berbagai fitur utama yang diperlukan agar sistem dapat menjalankan proses deteksi *phishing* sesuai dengan tujuan penelitian. Rincian kebutuhan fungsional tersebut dapat dilihat pada Tabel I.

TABEL I
HASIL ANALISIS KEBUTUHAN FUNGSIONAL

Kebutuhan	Deskripsi
Input URL	Sistem menyediakan kolom berupa *text box* untuk *user* memasukan URL yang akan diuji.
Validasi URL	Sistem memberikan kesempatan untuk *user* memeriksa ulang URL secara manual yang dimasukan sebelum diproses.
Ekstraksi Fitur	Setelah diinputkan URL, sistem secara otomatis melakukan ekstraksi fitur URL untuk mengidentifikasi pola dari URL yang telah diinputkan ke sistem.
Proses Klasifikasi	Sistem melakukan proses klasifikasi menggunakan model terbaik yang terpilih dari hasil evaluasi model.
Tampilkan Hasil	Sistem dapat menampilkan hasil klasifikasi akhir berdasarkan hasil model terbaik dalam bentuk tulisan *phishing* atau *legitimate* dan presentase indikasi label tersebut.
Input Ulang	Sistem akan secara otomatis mereset *input*, sehingga pengguna dapat langsung melakukan pengecekan terhadap tautan baru tanpa perlu melakukan pengaturan ulang secara manual.

2) Hasil Analisis Kebutuhan Non-Fungsional: Kebutuhan non-fungsional pada penelitian ini mencakup berbagai aspek pendukung yang diperlukan agar sistem dapat beroperasi secara optimal, aman, dan nyaman digunakan. Rincian kebutuhan non-fungsional tersebut dapat dilihat pada Tabel II.

TABEL II
HASIL ANALISIS KEBUTUHAN NON-FUNGSIONAL

Kebutuhan	Deskripsi
Koneksi Internet	Sistem memerlukan koneksi internet yang stabil agar website dapat diakses dan proses deteksi URL berjalan dengan baik.
Format URL Valid	URL yang dimasukkan harus menggunakan format yang valid sesuai dengan struktur umum URL dan lengkap agar dapat diproses oleh sistem.
Kompatibilitas Browser	Setelah diinputkan URL, sistem secara otomatis melakukan ekstraksi fitur URL untuk mengidentifikasi pola dari URL yang telah diinputkan ke sistem
Waktu Respons	Sistem melakukan proses klasifikasi menggunakan model terbaik yang terpilih dari hasil evaluasi model.
Tampilkan Hasil	Sistem dapat menampilkan hasil klasifikasi akhir berdasarkan hasil model terbaik dalam bentuk tulisan *phishing* atau *legitimate* dan presentase indikasi label tersebut.
Input Ulang	Sistem akan secara otomatis mereset input, sehingga pengguna dapat langsung melakukan pengecekan terhadap tautan baru tanpa perlu melakukan pengaturan ulang secara manual.

3) Hasil Analisis Perangkat Penelitian: Hasil analisis perangkat penelitian mencakup kebutuhan perangkat lunak dan perangkat keras yang digunakan untuk mendukung proses pengembangan, pelatihan model, serta implementasi sistem. Rincian perangkat yang digunakan dapat dilihat pada Tabel III.

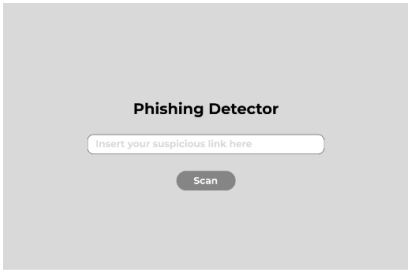
TABEL III
HASIL ANALISIS PERANGKAT PENELITIAN

Perangkat Lunak	Perangkat Keras
Online Web Browser	RTX 3050
Google Colab	512 GB SSD
Python Versi 3.12.2	Prosesor AMD Ryzen 5 (Intel Core i5)
Website Server Localhost	8 GB RAM
Visual Studio Code	Mouse Logitech
Framework Web Flask	GPU NVIDIA GeForce

B. User Design

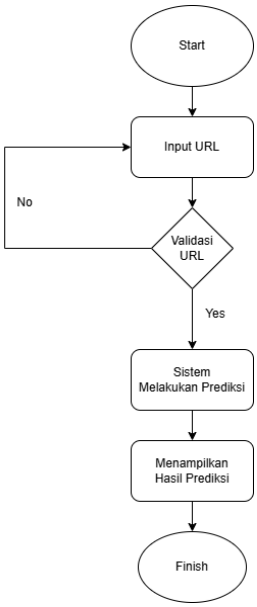
Pada tahap *user design*, dilakukan perancangan antarmuka dan alur kerja sistem sebagai gambaran awal implementasi. Hasil dari tahap ini meliputi *wireframe* antarmuka pengguna dan *flowchart* alur sistem, yang digunakan sebagai acuan dalam proses pengembangan sistem deteksi *phishing*.

1) Perancangan Desain Antarmuka: Hasil perancangan desain antarmuka pada penelitian ini disajikan dalam bentuk *wireframe. Wireframe* digunakan untuk menggambarkan struktur, tata letak, serta elemen-elemen utama pada sistem sebelum tahap implementasi dilakukan. Tampilan *wireframe* untuk sistem dapat dilihat pada Gbr. 6.



Gbr. 6 Perancangan Desain Antarmuka

1) *Perancangan Alur Sistem*: Hasil perancangan alur sistem pada penelitian ini disajikan dalam bentuk *flowchart*. *Flowchart* digunakan untuk menggambarkan urutan proses dan aliran kerja sistem dalam mendeteksi URL *phishing*, mulai dari input URL hingga menampilkan hasil klasifikasi. Tampilan *flowchart* dapat dilihat pada Gbr. 7.

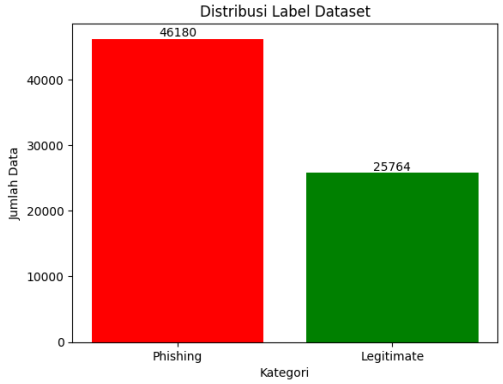


Gbr. 7 *Flowchart* Perancangan Sistem Klasifikasi Deteksi

C. Construction

Tahap *construction* terfokus pada proses pembuatan model dan sistem. Hal ini bisa dilalui dengan tahap pengumpulan dataset, pra pemrosesan data, seleksi fitur, *hyperparameter tuning*, evaluasi dan komparasi model, validasi data eksternal, dan integrasi model ke sistem.

1) Persiapan Dataset: Dataset yang digunakan pada penelitian ini terdiri atas dua jenis, yaitu dataset *phishing* dan dataset *legitimate*. Dataset *phishing* merupakan data sekunder yang diperoleh dari *PhishTank* berupa 46.181 URL *phishing* yang masih aktif dan terverifikasi, diambil pada tanggal 2 Januari 2026. Sementara itu, dataset *legitimate* merupakan data primer yang diperoleh melalui proses *crawling* terhadap 20.722 domain aman yang bersumber dari *Majestic Million*. Dari proses *crawling* tersebut dihasilkan sebanyak 25.765 URL *legitimate* yang aktif. Pada proses ini, hanya atribut Domain yang digunakan sebagai masukan untuk *crawling*, sedangkan atribut lainnya tidak digunakan karena tidak berpengaruh dalam proses klasifikasi. Dengan demikian, total dataset awal yang digunakan dalam penelitian ini berjumlah 71.946 URL, yang terdiri atas 46.181 URL *phishing* dan 25.765 URL *legitimate*. Untuk ilustrasi perbandingan yang lebih jelas dari kedua dataset dapat dilihat pada Gbr. 8.



Gbr. 8 Jumlah Awal Dataset *Phishing* dan *Legitimate*

2) *Pra Pemrosesan Data:* Tahapan ini dilakukan untuk memastikan data yang digunakan dalam proses analisis memiliki kualitas yang baik dan relevan untuk penelitian. Berikut rincian tahap pra pemrosesan data:

- *Attribute Reduction*: Pada tahap ini, kedua dataset disederhanakan sehingga hanya menyisakan atribut url sebagai data yang digunakan dalam penelitian. URL yang berasal dari hasil *crawling* domain *Majestic Million* akan digunakan untuk label *legitimate*, sedangkan URL yang diperoleh dari *PhishTank* akan digunakan untuk label *phishing*.

- *Data Labeling*: Pada tahap ini, dataset akan terbagi menjadi dua kelas utama, yaitu: *phishing* dan *legitimate*. Pelabelan ini menjadi dasar penting dalam pembentukan data latih dan data uji, sehingga model dapat mempelajari perbedaan karakteristik antara tautan *phishing* dan tautan *legitimate* secara lebih terstruktur.

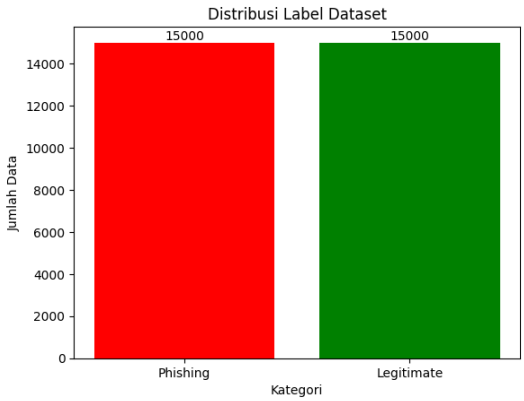
- *Data Cleaning*: Pada tahap *data cleaning* dilakukan normalisasi struktur URL dengan menghapus tanda *slash* di akhir URL, penghapusan data duplikat, serta penghapusan

URL yang tidak valid, seperti URL yang tidak dapat diakses, memiliki status *code* selain 200 *(active)*, bukan bertipe HTML, atau gagal diproses saat ekstraksi. Tahapan ini bertujuan menghasilkan dataset yang lebih bersih, konsisten, dan siap digunakan pada proses ekstraksi fitur. Setelah proses *data cleaning* selesai, sebanyak 78 URL duplikat dan 36.960 URL tidak valid dihapus, sehingga diperoleh 17.679 URL *phishing* dan 17.228 URL *legitimate* yang digunakan pada penelitian.

• *Ekstraksi Fitur*: Pada tahap ini akan dilakukan proses pengubahan data mentah yang ditambahkan hasil sekumpulan fitur yang bersifat representatif dan informatif. Adapun fitur yang digunakan adalah 55 fitur berbasis struktur URL *(URL-Based)* dan 20 fitur berbasis konten dalam URL *(Content-Based)*. *URL-Based Features* adalah fitur yang diekstraksi langsung berdasarkan struktur dan karakteristik URL tanpa perlu mengakses isi halaman website. Sedangkan *Content Based Features* adalah fitur yang diperoleh dari mendeteksi isi atau konten halaman website setelah URL diakses.

• *Data Merging*: Pada tahap ini, seluruh dataset dari label *phishing* dan *legitimate* akan digabung menjadi satu kesatuan dataset, sehingga memiliki karakteristik yang beragam.

• *Data Balancing*: Pada tahap ini akan dilakukan proses menyeimbangkan jumlah data pada setiap kelas agar tidak terjadi ketimpangan antara kelas *phishing* dan *legitimate*. Pada penelitian ini digunakan metode *undersampling* untuk menyamakan jumlah data pada kedua kelas sehingga model dapat mempelajari karakteristik masing-masing kelas dengan lebih seimbang dan menghasilkan performa klasifikasi yang lebih baik. Jumlah dataset setelah melakukan *data balancing* dapat dilihat pada Gbr. 9.



Gbr. 9 Dataset Phishing dan Legitimate Setelah Data Balancing

• *Data Shuffling*: Pada tahap ini akan dilakukan proses pengacakan susunan data dalam dataset sebelum proses pelatihan model, dengan tujuan untuk menghilangkan pengaruh urutan data asli.

3) Seleksi Fitur Chi-Square: Pada penelitian ini, *Chi-Square* berfungsi untuk memberikan ranking pada fitur untuk mengetahui relevannya setiap fitur untuk penentuan label klasifikasi. Berikut proses perhitungan mengggunakan *Chi-Square:*

• *Jumlah Data Tiap Label*: Berdasarkan hasil dari akhir preprocessing, diperoleh dataset dengan total 30.000 data yang terdiri dari dua label: 15.000 dataset *phishing* dan 15.000 dataset *legitimate.*

• *Observed Value (O)*: Merupakan nilai yang diperoleh langsung dari data dan digunakan untuk menunjukkan distribusi aktual suatu fitur terhadap masing-masing kelas. Pada penelitian ini, nilai *observed value* dihitung dengan menjumlahkan seluruh nilai fitur pada setiap kelas. Sebagai contoh, pada atribut num_hyperlinks diperoleh *observed value* sebesar 288.768 untuk kelas *phishing* dan 3.361.980 untuk kelas *legitimate*, dengan total keseluruhan sebesar 3.650.748.

• *Proporsi Kelas*: Menunjukkan perbandingan jumlah data pada setiap kelas terhadap total dataset. Pada penelitian ini, setelah proses *data balancing*, jumlah data *phishing* dan *legitimate* masing-masing sebanyak 15.000 data sehingga memiliki proporsi yang sama, yaitu 0,5. Dengan demikian, dataset memiliki distribusi kelas yang seimbang dan tidak terdapat ketimpangan jumlah data antar kelas.

• *Expected Value (E)*: Merupakan nilai frekuensi yang diharapkan apabila distribusi nilai fitur terhadap kelas mengikuti proporsi jumlah data pada masing-masing kelas. Nilai ini digunakan sebagai pembanding terhadap *observed value (O)* dalam perhitungan statistik *Chi-Square*. Pada penelitian ini, expected value dihitung menggunakan rumus total nilai fitur × proporsi kelas. Karena proporsi kelas *phishing* dan *legitimate* seimbang, yaitu 0,5 : 0,5, maka masing-masing kelas memiliki *expected value* sebesar 1.825.374. Hasil tersebut menunjukkan bahwa distribusi harapan fitur num_hyperlinks pada kedua kelas adalah sama sesuai dengan proporsi data yang digunakan.

• *Chi-Square*: Nilai *Chi-Square* dihitung dengan tujuan untuk mengetahui perbedaan antara *observed value (O)* dan *expected value (E).* Seperti contoh, akan dihitung nilai *Chi* pada atribut num_hyperlinks. Perhitungan perbedaan antara *observed value (O)* dan *expected value (E)* dilakukan menggunakan persamaan rumus *Chi-Square* pada persamaan (1).

$$\chi^2 = \sum \frac{(O-E)^2}{E} \quad (1)$$

Dari tahap sebelumnya, telah dihitung *observed value (O)* dan *expected value (E)* dengan hasil sebagai berikut:

Observed value:

O_1 *(phishing)* = 240932 dan O_2 *(legitimate)* = 2751950

Expected value:

E_1 *(phishing)* = 2992882 dan E_2 *(legitimate)* = 2992882

Jika semua nilai sudah ada, saatnya mengaplikasikan ke rumus *Chi-Square* seperti pada persamaan (2) untuk mendapatkan hasil akhirnya:

$$\chi^2 = \frac{(O_1 - E_1)^2}{E_1} + \frac{(O_2 - E_2)^2}{E_2} \quad (2)$$

Sehingga akan didapatkan hasil akhir nilai *Chi-Square* pada atribut num_hyperlinks sebesar χ^2 = 2587040,24.

Kesimpulannya, nilai *Chi-Square* sebesar 2.587.040,24 menunjukkan adanya perbedaan yang besar antara *observed value (O)* dan *expected value (E),* sehingga fitur tersebut memiliki hubungan yang kuat terhadap label klasifikasi. Seluruh fitur akan dihitung nilai *Chi*-nya. Semakin besar nilai *Chi-Square* yang diperoleh, maka semakin besar pula kontribusi fitur dalam membedakan kelas *phishing* dan *legitimate*, sehingga fitur tersebut dianggap relevan dan layak dipertimbangkan dalam proses seleksi fitur.

Seleksi fitur akan dilakukan dengan rentang 15 hingga 35 fitur, nantinya fitur-fitur tersebut akan dilakukan *data splitting* untuk melihat keefektifannya. Hasil seleksi fitur juga akan dibandingkan dengan data yang tidak menggunakan tahap seleksi fitur.

4) Hyperparameter Tuning: Pada tahap ini dilakukan tuning hyperparameter algoritma *Random Forest* menggunakan metode *GridSearchCV* untuk mencari kombinasi hyperparameter terbaik. Daftar hyperparameter yang akan dilakukan *tuning* tertera pada Tabel IV.

TABEL IV
DAFTAR HYPERPARAMETER ALGORITMA RANDOM FOREST

Hyperparameter	Nilai
n_estimators	100, 200, 300
max_depth	None, 10, 20
min_samples_split	2, 5
min_samples_leaf	1, 2
max_features	sqrt

Dari beberapa hyperparameter di atas, setelah melakukan proses *tuning* dengan *GridSearchCV* maka akan menghasilkan hyperparameter terbaik yang nantinya akan digunakan dalam model. Hasil dari *tuning* dan hyperparameter terbaik dapat ditinjau pada Tabel V.

TABEL V
HYPERPARAMETER TERBAIK ALGORITMA RANDOM FOREST

Hyperparameter	Nilai
n_estimators	300
max_depth	None
min_samples_split	2
min_samples_leaf	1
max_features	sqrt

Tabel V merupakan hasil dari proses GridSearchCV yang menunjukkan kombinasi hyperparameter terbaik pada algoritma *Random Forest* berdasarkan nilai akurasi sebesar 99.67% (0.9966666666667) dan nilai tersebut adalah hasil akurasi tertinggi yang diperoleh selama proses pengujian.

5) Evaluasi dan Komparasi Model Klasifikasi: Pada tahap evaluasi dan komparasi model akan dilakukan metode *data splitting*, yaitu membagi dataset menjadi dua bagian, data latih *(training set)* dan data uji *(testing set)* menggunakan lima skenario, yaitu: 90:10, 80:20, 70:30, 60:40, 50:50. Pada prosesnya akan menghasilkan *classification report*. Dalam tahap ini juga akan diambil model dengan hasil terbaik dari segi hyperparameter dan jumlah fitur yang terseleksi untuk diintegrasikan ke sistem.

Berdasarkan hasil pengujian pada berbagai skenario pembagian data, akan diambil satu hasil terbaik. Hasil terbaik didapat pada skenario 90:10 dengan 33 fitur mendapatkan hasil akurasi 0.998 menggunakan algoritma *Random Forest*. Model inilah nantinya yang akan diintegrasikan ke sistem. Adapun 33 fitur tersebut dapat ditinjau pada Tabel VI.

TABEL VI
HASIL SKENARIO TERBAIK DARI EVALUASI MODEL

No	Nilai Chi2-Score	Fitur	Deskripsi
1	2.587040e+06	num_hyperlinks	Jumlah seluruh hyperlink pada halaman website. Hal tersebut mencakup tag <a>, <link>, <media>, <form>, <css>, <favicon>.
2	4.360629e+05	internal_media_ratio	Presentase media yang meliputi foto, audio, dan video terhadap total media dari domain internal.
3	5.262216e+04	longest_word_url	Jumlah karakter dalam sebuah kata terpanjang dalam URL.
4	3.745713e+04	external_media_ratio	Presentase media yang meliputi foto, audio, dan video terhadap total media dari domain eksternal.
5	2.583928e+04	unsafe_anchor	Presentase anchor (<a>) berbahaya seperti javascript,

			mailto, dan anchor yang kosong.
6	1.967457e+04	hostname_length	Panjang karakter hostname dalam sebuah URL. Hostname terdiri dari subdomain, domain, dan top-level domain.
7	1.926109e+04	url_length	Panjang karakter URL keseluruhan.
8	1.143079e+04	longest_word_path	Jumlah karakter terpendek di path dalam URL.
9	6.984932e+03	count_hyphen	Jumlah karakter – dalam URL
10	5.712369e+03	count_equal	Jumlah karakter = dalam URL.
11	4.085302e+03	count_ampersand	Jumlah karakter & dalam URL.
12	3.787369e+03	char_repeat	Jumlah karakter yang berulang dalam URL.
13	3.671000e+03	suspecious_tld	Terdapat TLD mencurigakan dalam URL.
14	3.251000e+03	count_semicolon	Jumlah karakter ; dalam URL.
15	3.228354e+03	count_dot	Jumlah karakter . dalam URL.
16	3.228354e+03	subdomain_count	Jumlah subdomain dalam URL.
17	2.932388e+03	internal_link_ratio	Rasio jumlah link internal dibandingkan dengan total hyperlink.
18	2.368761e+03	invisible_iframe	Terdapat iframe dengan ukuran yang tidak terlihat.
19	2.268660e+03	domain_in_brand	Penggunaan nama brand di domain dalam URL.
20	2.202319e+03	empty_title	Judul pada halaman web kosong.
21	1.838334e+03	random_domain	Penggunaan domain acak pada URL.
22	1.723107e+03	longest_word_host	Jumlah karakter terpanjang di hostname dalam URL.
23	1.708762e+03	count_question	Jumlah karakter ? dalam URL.
24	1.250051e+03	shortening_service	Penggunaan shortener link dalam URL.
25	1.238638e+03	count_underscore	Jumlah karakter _ dalam URL.
26	1.022125e+03	login_form_abnormal	From login yang indikasinya mengarah ke domain eksternal atau kosong.
27	1.022125e+03	sfh	Terdapat aksi form kosong atau domain eksternal.
28	8.486432e+02	shortest_word_url	Jumlah karakter dalam sebuah kata terpendek dalam URL.
29	7.645398e+02	count_percent	Jumlah karakter % dalam URL. Nilainya berupa bilangan bulat.
30	6.906144e+02	count_slash	Jumlah karakter / dalam URL. Nilainya berupa bilangan bulat.
31	6.897944e+02	shortest_word_path	Jumlah karakter terpendek di path dalam URL.
32	6.396369e+02	count_com	Jumlah string com dalam URL.
33	6.161347e+02	shortest_word_host	Jumlah karakter terpendek di hostname dalam URL.

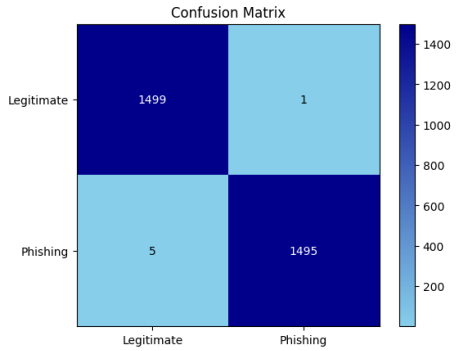
Fitur-fitur yang didapat telah diurutkan berdasarkan peringkat *Chi2-Score*. Selanjutnya pada Tabel VII menunjukan *classification report* dari skenario *data splitting* 90:10 dengan akurasi sebesar 0,998 dan melakukan proses klasifikasi menggunakan algoritma *Random Forest* dengan 33 fitur.

TABEL VII
CLASSIFICATION REPORT RANDOM FOREST 90:10 (33 FITUR)

Skenario	Label	Precision	Recall	F1-Score	Support
90:10	Phishing	0.99933	0.99666	0.99799	1500
	Legitimate	0.99667	0.99933	0.99800	1500
	Macro Avg	0.99800	0.99800	0.99800	3000

	Weighted Avg	0.99800	0.99800	0.99800	3000

Berdasarkan hasil evaluasi pada skenario 90:10 dengan 33 fitur ini didapat dari *confusion matrix* pada Gbr. 10.



Gbr. 10 *Confussion Matrix Random Forest* 90:10 (33 Fitur)

Berdasarkan *confusion matrix* pada Gbr. 10, dari total 3000 data uji, sebanyak 1499 data *legitimate* dan 1495 data *phishing* berhasil diklasifikasikan dengan benar. Kesalahan klasifikasi kecil ada pada 1 data *legitimate* yang diprediksi sebagai *phishing* (FP) dan 5 data *phishing* yang diprediksi sebagai *legitimate* (FN).

Untuk perhitungan hubungan antara *classification report* dan *confussion matrix* dalam hasil evaluasi ini adalah sebagai berikut:

- Accuracy

$$Accuracy = \frac{1495 + 1499}{1495 + 1499 + 1 + 5} = 0{,}998 = 99{,}8\%$$

- Precision

$$Precision\ (Phishing) = \frac{1495}{1495 + 1} = 0{,}99933 = 99{,}93\%$$

$$Precision\ (Legitimate) = \frac{1499}{1499 + 5} = 0{,}99667 = 99{,}66\%$$

- Recall

$$Recall\ (Phishing) = \frac{1495}{1495 + 5} = 0{,}99666 = 99{,}66\%$$

$$Recall\ (Legitimate) = \frac{1499}{1499 + 1} = 0{,}99933 = 99{,}93\%$$

- F1-Score

$$F1 - Score\ (Phishing) = 2 \times \frac{(0{,}99933 \times 0{,}99666)}{(0{,}99933 + 0{,}99666)} = 0{,}99799$$

$$F1 - Score\ (Legitimate) = 2 \times \frac{(0{,}99667 \times 0{,}99933)}{(0{,}99667 + 0{,}99933)} = 0{,}99800$$

- Macro Average

$$Macro\ Precision = \frac{0{,}99933 + 0{,}99667}{2} = 0{,}99800 = 99{,}8\%$$

$$Macro\ Recall = \frac{0{,}99666 + 0{,}99933}{2} = 0{,}99800 = 99{,}8\%$$

$$Macro\ F1 - Score = \frac{0{,}997997 + 0{,}999333}{2} = 0{,}998000 = 99{,}8\%$$

- Weighted Average

$$Weighted\ Precision = \frac{(1500 \times 0{,}99933 + 1500 \times 0{,}99667)}{3000} = 0{,}99800$$

$$Weighted\ Recall = \frac{(1500 \times 0{,}99666 + 1500 \times 0{,}99933)}{3000} = 0{,}998000$$

$$Weighted\ F1 - Score = \frac{(1500 \times 0{,}99799 + 1500 \times 0{,}99800)}{3000} = 0{,}99800$$

Dengan ini dapat disimpulkan bahwa hasil akurasi akhir dari model *Random Forest* 33 fitur pada skenario 90:10 mendapatkan akurasi tebesar yaitu senilai 99,8%.

6) Validasi Data Eksternal: Pengujian validasi data eksternal dilakukan untuk mengevaluasi kinerja model dalam mendeteksi URL yang berasal dari luar dataset pelatihan maupun dataset pengujian sebelumnya. Pada tahap ini, digunakan sebanyak 10 URL. Seluruh URL tersebut dipilih dari sumber eksternal sehingga tidak pernah digunakan pada proses pelatihan maupun evaluasi model sebelumnya. Hasil validasi data eksternal dapat dilihat pada Tabel VIII.

TABEL VIII
HASIL VALIDASI DATA EKSTERNAL

No	URL	Label Asli	Hasil Prediksi
1	https://github.com/topics/machine-learning	Legitimate	Legitimate
2	https://my.1passvvord.org/verify	Phishing	Phishing
3	http://smsenligne.myfreesites.net	Phishing	Phishing
4	http://www.plaza-surabaya.com/tenant/79	Legitimate	Legitimate
5	https://www.netflix.com/title/80057281	Legitimate	Legitimate
6	https://secure-domainrenew.com/at-arr/	Phishing	Phishing
7	http://bit.ly/udmcmadCracked	Phishing	Phishing
8	https://bit.ly/4mFX5eL	Phishing	Phishing
9	https://bit.ly/48Ywtjt	Legitimate	Phishing
10	https://www.apple.com/id/iphone	Legitimate	Legitimate

7) *Implementasi Sistem:* Pada tahap ini, implementasi sistem dilakukan melalui beberapa tahapan utama, meliputi perancangan arsitektur sistem, perancangan tampilan antarmuka, serta implementasi komponen *front-end*, *back-end*, dan model klasifikasi yang digunakan.

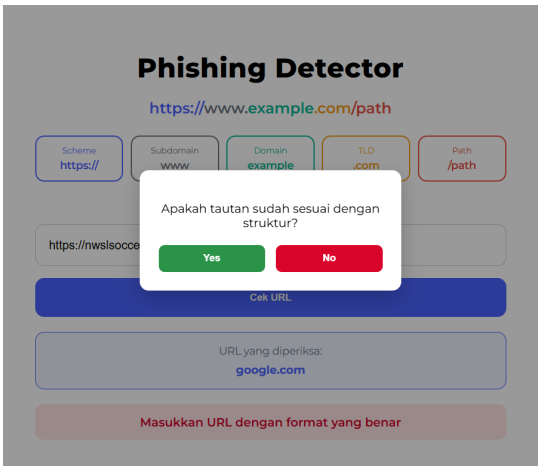
• *Arsitektur Sistem:* Arsitektur sistem pada penelitian ini dirancang untuk mendukung implementasi sistem klasifikasi deteksi tautan berbasis web yang dijalankan secara lokal. Perancangan arsitektur difokuskan pada pemisahan fungsi antar komponen agar proses validasi URL, ekstraksi fitur, klasifikasi, dan penyajian hasil dapat berjalan secara terstruktur, efisien, serta mudah dikembangkan.

• *Web Browser*: Digunakan sebagai media untuk mengakses sistem deteksi tautan *phishing*, sebab sistem yang dihasilkan berbasis website. Melalui web browser, pengguna dapat mengakses *front-end* sistem yang berfungsi sebagai antarmuka utama untuk berinteraksi dengan aplikasi.

• *Frontend:* Antarmuka pengguna yang diakses melalui web browser untuk berinteraksi dengan sistem deteksi *phishing*. Pada sistem ini, *front-end* dibangun menggunakan bahasa pemrograman HTML dan CSS untuk menghasilkan tampilan antarmuka yang responsif, menarik, dan mudah digunakan. Tampilan awal antarmuka sistem dapat dilihat pada Gbr. 11.

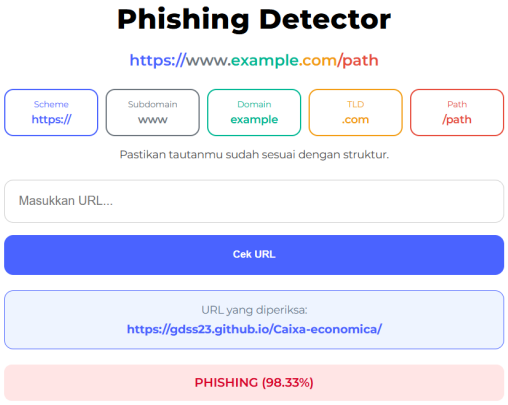
Gbr. 11 Tampilan Awal Antarmuka Sistem

Selanjutnya menuju ke tampilan kedua yaitu tampilan *pop-up* validasi manual untuk *user*. Tampilan validasi manual untuk *user* dapat dilihat pada Gbr. 12.



Gbr. 12 Tampilan Validasi Manual untuk *User*

Kemudian yang terakhir yaitu tampilan ketiga akan menunjukan tampilan akhir sistem setelah keluar hasil prediksi deteksinya. Tampilan akhir antarmuka sistem dapat dilihat pada Gbr. 13.

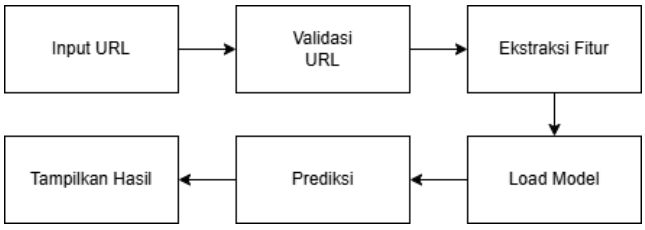


Gbr. 13 Tampilan Akhir Antarmuka Sistem

• *Backend: Back-end* aplikasi dibangun menggunakan *framework Flask* dan berperan sebagai pusat pengelolaan logika sistem. *Back-end* bertugas menangani seluruh proses utama mulai dari penerimaan input hingga pengiriman hasil klasifikasi kepada pengguna.

• *Model Prediksi*: Model prediksi yang digunakan pada penelitian ini telah dilatih sebelumnya menggunakan dataset *phishing* dan *legitimate*. Model disimpan dalam format *Pickle (.pkl)* dan dimuat ke dalam sistem menggunakan *library Joblib.*

8) *Implementasi dan Integrasi Model ke Sistem*: Pada tahap ini, model terbaik yang diperoleh dari hasil pengujian diintegrasikan ke dalam sistem deteksi *phishing* berbasis web. Model tersebut adalah model dengan hasil seleksi *Chi-Square* serbanyak 33 fitur menggunakan algoritma *Random Forest* sebagai model klasifikasi. Model tersebut disimpan dalam file berekstensi *.pkl* menggunakan *library Joblib*, sehingga dapat digunakan kembali tanpa perlu melakukan proses pelatihan ulang. Implementasi sistem dilakukan menggunakan *framework Flask* sebagai *backend* yang berfungsi untuk menghubungkan antarmuka pengguna dengan model *machine learning.* Alur implementasi dan integrasi model ke sistem dapat ditinjau pada Gbr. 14.



Gbr. 14 Alur Implementasi Model ke Sistem

D. Cutover

Tahap *cutover* berfokus pada proses pengujian akhir sistem sebelum digunakan sepenuhnya. Pada tahap ini, kegiatan yang dilakukan adalah pengujian fungsional sistem. Hasil pengujian *blackbox testing* dapat dilihat pada Tabel IX.

TABEL IX
HASIL VALIDASI DATA EKSTERNAL

Skenario	Atribut	Ekspetasi Output	Hasil
Input URL Phishing	Tautan berbahaya yang terverifikasi aktif.	Sistem menampilkan hasil deteksi berupa phishing beserta presentase probabilitasnya.	Sesuai
Input URL Legitimate	Tautan aman yang terverifikasi aktif.	Sistem menampilkan hasil deteksi berupa legitimate beserta presentase probabilitasnya.	Sesuai
Input teks random	Teks random dengan spasi dan tidak berstruktur URL	Sistem menolak input	Sesuai
Input URL tidak berstruktur HTML	Tautan yang konten websitenya tidak menggunakan HTML	Sistem menolak input	Sesuai
Input Kosongan	Tidak melakukan input teks ataутautan, namun langsung meminta sistem mengecek.	Sistem menolak input	Sesuai

V. KESIMPULAN

Penelitian ini berhasil mengekstraksi 75 fitur URL, kemudian menyeleksi 33 fitur terbaik menggunakan metode *Chi-Square*. Model *Random Forest* memberikan performa terbaik dengan *accuracy* 99,8% (0,998). *Accuracy* tersebut didapatkan dari hasil *classification report* dengan *precision phishing* sebesar 0.999332 dan *legitimate* sebesar 0.996676, *recall phishing* sebesar 0.996667 dan *legitimate* sebesar 0.999333, serta *F1-Score phishing* sebesar 0.997997 dan *legitimate* sebesar 0.998003 pada skenario pembagian data 90:10. Model tersebut selanjutnya berhasil diimplementasikan ke dalam sistem berbasis website menggunakan metode RAD untuk mendeteksi tautan *phishing* secara *real-time.*

REFERENSI

[1] V. A. Windarni *et al.*, "DETEKSI WEBSITE PHISHING MENGGUNAKAN TEKNIK FILTER PADA MODEL MACHINE LEARNING Abstraksi Kata Kunci : Decision Tree , Naïve Bayes , Random Forest , Website Phishing Keywords : Decision Tree , Naïve Bayes , Random Forest , Website Phishing Pendahuluan Meto," vol. 6, no. 1, pp. 39–43, 2023.

[2] A. D. Harahap, D. Juardi, and A. S. Y. Irawan, "Rancang Bangun Sistem Pendeteksi Link Phishing Menggunakan Algoritma Random Forest Berbasis Web," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, 2024, doi: 10.23960/jitet.v12i3.4858.

[3] P. Chyan *et al.*, *Pengantar Machine Learning PT. MIFANDI MANDIRI DIGITAL*, vol. 1. 2024. [Online]. Available: http://jurnal.mifandimandiri.com/index.php/penerbitmmd/article/view/38

[4] T. Sistem and M. Fahri, "Informasi," vol. 6, no. 2, pp. 186–194, 2025, doi: 10.62527/jitsi.6.2.472.

[5] A. A. Nopebrian, R. N. Shofa, and S. Yuliyanti, "86639-75676751783-1-Pb," vol. 13, no. 1, pp. 166–172, 2025, doi: 10.26418/justin.v13i1.86639.

[6] Y. H. Lokapala, F. J. Nurfauzi, and Y. Widowaty, "Aspek Yuridis Kejahatan Phishing dalam Ketentuan Hukum di Indonesia," *Indones. J. Crim. Law Criminol.*, vol. 5, no. 1, pp. 19–24, 2024, doi: 10.18196/ijclc.v5i1.19853.

[7] C. Do Xuan, H. D. Nguyen, and T. V. Nikolaevich, "Malicious URL detection based on machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 1, pp. 148–153, 2020, doi: 10.14569/ijacsa.2020.0110119.

[8] R. R. Adhitya, Wina Witanti, and Rezki Yuniarti, "Perbandingan Metode Cart Dan Naïve Bayes Untuk Klasifikasi Customer Churn," *INFOTECH J.*, vol. 9, no. 2, pp. 307–318, 2023, doi: 10.31949/infotech.v9i2.5641.

[9] E. Hokijuliandy and H. Napitupulu, "Application of SVM and Chi-Square Feature Selection for Sentiment Analysis of Indonesia ' s National Health Insurance Mobile Application," 2023.

[10] R. Z. Zakiyah and M. A. Islam, "User Interface Website Sebagai Media Promosi Vilovy Design," *J. Barik*, vol. 3, no. 3, pp. 174–185, 2022, [Online]. Available: https://ejournal.unesa.ac.id/index.php/JDKV/article/download/48070/40140/

[11] D. Hariyanto *et al.*, "Implementasi Metode," *J. Al-ilmi*, vol. 13, no. 1, pp. 110–117, 2021.

[12] C. C. Siburian, A. Andriani, and C. B. Dewa, "Aplikasi Mypresent Untuk Pengelolaan Data Presensi Karyawan Dengan Metode Rad," *J. Inf. Syst. Manag.*, vol. 5, no. 2, pp. 219–226, 2024, doi: 10.24076/joism.2024v5i2.1435.

[13] N. Made Dwi Febriyanti, A. A. Kompiang Oka Sudana, and I. Nyoman Piarsa, "Implementasi Black Box Testing pada Sistem Informasi Manajemen

Dosen," *Jitter- J. Ilm. Teknol. dan Komput.*, vol. 2, no. 3, pp. 1–10, 2021.

[14] R. Ilyasa, A. Lelitasari, R. G. Satria, and R. A. Vetian, "APLIKASI FORUM DISKUSI DAN BELAJAR MANDIRI UNTUK SISWA MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT (RAD)," vol. 9, no. 1, pp. 71–80, 2024.

[15] I. Roery, S. L. Mufreni, and E. P. Silmina, "Sistem informasi klinik berbasis web dengan menggunakan metode Rapid Application Development (RAD) Web-based clinic information system using Rapid Application Development (RAD) method," vol. 3, pp. 76–87, 2025.